2

ARMY RESEARCH LABORATORY

Assessment of Projectile Shatter Using Digital Image Processing Techniques

James D. Kleinmeyer

ARL-TR-92

April 1993

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1993	Final Report

4. TITLE AND SUBTITLE

Assessment of Projectile Shatter Using Digital Image Processing Techniques

5. FUNDING NUMBERS

6. AUTHOR(S)

James D. Kleinmeyer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
Watertown, Massachusetts 02172-0001
ATTN: AMSRL-MA-PB

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-92

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, Maryland 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes a novel technique for identifying the advent of projectile shatter in ballistically tested armor specimens. The methodology proposed is based on the use of digital image processing techniques for visually mapping out the interior of a ballistic cavity. Such a map may then be used to determine if projectile shatter has occurred and if so, to gain additional information on the approximate depth, location, and degree of shatter. The implementation of this technique will result in improved accuracy and efficiency in the analysis of projectile shatter. The first and most noticeable savings is in eliminating the requirement for machining and preparation of specimens. This will provide for a quicker turn around time, resulting either in increased labor savings or increased sample throughput. Since the samples need not be machined, the penetration cavity remains intact in its original state, which, upon analysis, should result in a more accurate account of the projectile/armor interaction. An additional benefit is that the system may be easily automated for data acquisition,allowing for an additional increase in sample processing. The recording of images by the host computer offers greater flexibility for study by the researcher. Since the images are archived using magnetic media, they may be recalled at any time to provide much of the same information as the original sample and in a much more convenient format. As techniques for image processing and armor analysis improve these same images may be reprocessed using the newer techniques, providing additional data without the need for sample and testing redundancy.

14. SUBJECT TERMS

Real time, Image processing, Damage, Armor, Test method

15. NUMBER OF PAGES 25

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Contents

I. Introduction 1

II. Image Processing Techniques. 1

III. Experimental Procedure 3

IV. Program Savings. 5

V. Results and Discussion 6

VI. Conclusions and Recommendations. 12

VII. Appendix A 13

Figures

1. System block diagram 4

2. Exterior view of armor sample. 7

3. Exterior view of armor sample (3D) 8

4. Interior cavity of ballistic sample. 9

5. Interior cavity of ballistic sample (high resolution). 10

6. Interior view of armor sample. 11

I. Introduction

Since the military specifications which dictate the conditions under which armor is either accepted or rejected often require ballistic testing, the proper selection of the appropriate armor thickness, obliquity, and type of kinetic energy projectile are critical. Misleading data may result from a test which is performed with a target that does not correctly represent the armor's performance e.g., when target conditions are such that incipient projectile shatter occurs. Because the beginning of projectile shatter is usually accompanied by a wide variation in projectile performance and is therefore dependent on the integrity of the projectile, target conditions which are designed to measure armor response should avoid the shatter regime and therefore not be included in military specifications. The presence of projectile shatter is usually defined by a detectable increase in the size of the penetration cavity accompanied by an increase in cavity interior roughness.

This report describes a novel technique for identifying the advent of projectile shatter. The methodology proposed is based on the use of digital image processing techniques for visually mapping out the interior of a ballistic penetration cavity. Such a map may then be used by experts to determine if projectile shatter has occurred and if so, possibly gain additional information as to the approximate depth, location and degree of projectile shatter as well.

II. Image Processing Techniques

In general, an image processor works by digitizing an incoming analog signal sent from a camera. An incoming image is divided into a grid of "pixels" (picture elements) displayed as the image on a monitor. Each pixel has a value in memory ranging from zero to 255 with zero corresponding to pure black and 255 corresponding to pure white. These gray scale values are placed in array locations in computer memory corresponding to their positions in the image. Contrast plays an important role in image processing since the sharper the contrast achieved, the greater the variation in the resulting pixel values. A critical factor in achieving good contrast is the lighting used. The image processor analyzes pixel values and manipulates images by changing the pixel values. These manipulations can take on several forms; digital filtering, lookup table manipulation and logical operation.

<u>Digital Filtering</u>: Digital filtering provides an effective method for the manipulation of images through the modification of each pixel relative to its neighbors. One such implementation of digital filtering is through the use of 2-dimensional convolution. This mathematical method is used to calculate a weighted average of each pixel based on the intensities of its neighbors. The digital filter is implemented by convolving the image matrix with a "kernel" matrix which is specified prior to the operation. The kernel is responsible for defining the coefficients needed to implement the desired filter.[1] For example, a 3x3 matrix kernel may be convolved with the image matrix

1. A.K. Jain, "Fundamentals of Digital Image Processing", Englewood Cliffs, NJ: Prentice Hall, 1989.

$$\begin{matrix} 0 & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & 0 \end{matrix}$$

where X is the target pixel and 0 is a pixel in an adjacent screen position. The filter action is dependent upon specification of the kernel coefficients. Different filtering techniques are used to produce specific results. For example, the matrix

$$\begin{matrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{matrix}$$

is used as a typical "high-pass" filter. This produces greater clarity and detail in the resulting image. Conversely, the matrix

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

functions as a "low-pass" filter which tends to blur details and give a better general outline of objects within the image than in the unprocessed image. This is extremely useful for artificial intelligence applications using object recognition.

In principle, an image processor is capable of handling any size matrix. Rectangular matrices, for example, are used for vertical and horizontal edge detection. Matrix convolution, however, requires many calculations such that the larger the kernel, the longer the filtering takes. A faster means of image manipulation relying on specific hardware within the system is often desirable for more efficient image analysis.

Lookup Tables: Lookup Tables, referred to as LUTs, are hardware mechanisms used to manipulate pixel values. Data can be passed through LUTs, which significantly accelerate the manipulative process. LUT operations are programmed to allow for flexibility in specifying their operation. Unlike digital filtering, LUTs affect only the target pixel without regard to its neighbors. As such, it is limited to changing pixel values by constant increments. The programming capability allows the increment to vary depending upon a pixel's original value. For example, an LUT may be programmed to invert an image - the image data is passed through the LUT and gray level values are inverted to give a negative image; i.e., zero values become 255, 1 becomes 254, etc. Since an image system may have as many as 24 LUTs, the flexibility in using these hardware registers is immense. LUTs can be programmed to set all but a limited range of values to zero, cause all values within a range to be set equal, or any combination of a number of LUT operations can be performed simultaneously using multiple LUTs.

Logical Operation: As mentioned previously, space is reserved in memory for as many as 24 LUTs to be present at one time. Likewise, space is reserved in memory for four different frame buffers, each having four image registers. Image registers are used to display the image on the monitor. By calling the frame buffer and register, the stored image is displayed. Multiple image registers allow not only for multiple images to be stored and viewed, but also provide the means for the use of logical operations. By applying the logical functions, images can be used to affect other images. In applying subtraction or addition, for example, features may be added or subtracted from the target image. The image registers can also display live data - a standard backdrop can be subtracted from the live image to allow for motion detection. Image registers do not modify pixel values. They are used instead as the storage location for the pixel values of the image they are used to display. Multiple registers, however, can be used to create new images from two or more registers.

III. Experimental Procedure

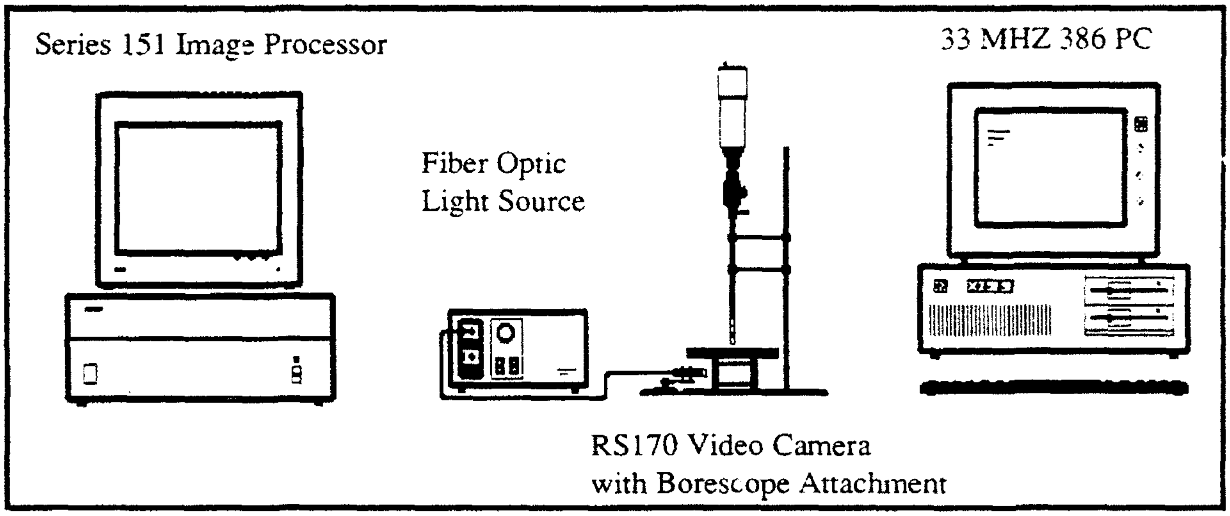
An Imaging Technologies Series 151 image processor coupled to a 33 MHZ 386 PC was used to acquire, process and store a series of consecutive images of the interior of each penetration cavity. An ITI (Instrument Technology Inc.) 9.2 mm diameter rigid mirror relay borescope was used in conjunction with a coupled RS170 camera to acquire images at fixed angles of rotation as well as varying depths as the borescope passed through the projectile cavity. The image processor was used to process the images in order to provide a detailed map of the cavity on an adjacent computer monitor. The map provides a visual reference similar to that of the contour maps of the earth provided by satellites. In this case images showing angular rotation are displayed along the x axis while images showing cavity depth are displayed along the y axis.

Once the operator has located a specific area of interest on the map (i.e. the point of projectile shatter). The appropriate images may then be called up by the computer for a detailed examination of the specific cavity area. If further enhancement of the image is desired, the image processor may than again be used to compensate for image deficiencies.

Lighting: The borescope/camera test fixture combination needs to be sufficiently flexible to allow for variations in sample thickness, cavity angle, and lighting, yet rigid enough in construction such that, once suitable settings were found, the fiberoptic lighting, borescope and camera could be locked into position to assure reliable operation. Because of the flexibility of the lighting system a number of techniques were used to illuminate the interior of the ballistic cavity. The borescope illumination system consisted of a dual range 150/1000 watt quartz halogen fiberoptic light source. By decoupling the fiberoptic light source from the borescope and using it as a detached point illumination source each cavity section could be illuminated. It is important to note here that as designed the borescope has the light source coupled at one end with a fiberoptic guide running almost the length of the probe and dispersing the light approximately 1.5 cm below the borescope window. For macroscopic (close up) work such as this it was found that the light could easily be scattered or completely blocked by the interior surface roughness of the cavity. By allowing the light to remain detached and flexible, it could be positioned either above or below the specimen allowing for adequate lighting during borescope positioning.

Camera Focal Length: To solve focal length problems brought about by the RS170 C mount adapter and borescope slip ring combination, a 20.5 mm C mount extension tube was added between the camera and the borescope adapter. By adding the extension tube set, the camera focal point was brought back to just outside the borescope window. This in turn allowed for a sharp macro lens like focus of the ballistic cavity wall by the borescope/camera combination. The camera could then be easily located on the light table directly above the sample.

Figure 1. System block diagram



Software Considerations: The Imaging Technology Series 151 Image Processor is designed to be programmed in the C programming language. Interpreter software included with the system was used to establish routines for isolating damage regions within each armor test specimen. Recognizing that the location of each damage region within the cavity is important to the operator a program to create a well defined map of the interior projectile cavity was written. This was originally written as three separate modules to capture, map and display 256 gray scale 512x512 images of the cavity interior at fixed angles of rotation and depth as the the borescope passed through the projectile cavity (See Appendix A). These routines were latter modified to handle the simultaneous capture, mapping and display of RGB color images in 3 (256 level) bit planes also at 512x512 resolution, yielding 24 bit RGB color digitization. For this system the images are acquired from a RGB Cohu Camera via three separate frame grabber inputs within the image processor. These images are then saved as separate 8 bit image files representing the Red, Green, and Blue bit planes. For display purposes the images are read into 3 separate frame buffers and displayed simultaneously via the Red, Green, and Blue LUTs to a RGB video monitor. Since the RGB software behaves identically to the B&W version with the exception of an additional 2 bit planes of imaging data, the RGB display program code was included only to provide completeness.

In addition to including RGB color capability, several routines are being scheduled for addition to help provide a general purpose imaging toolbox to the operator. Such tools would include both highpass and lowpass filters, gray scale enhancement through histogram equalization and stretch frame operations, and a window zoom feature for closer study of cavity interior. Provisions have also been made to accommodate for the generation of hardcopy output from the imaging monitor via either a wax thermal video printer or through a postscript interface to a laser printer.

IV. Program Savings

Such a technique promises to demonstrate substantial savings. The first and most noticeable is the lack of sample machining and preparation. The armor test samples are no-longer required to spend several hours (approx. 24 hrs./sample) in the machine shop being cut into smaller samples, with each penetration cavity being bisected for closer observation and study. In addition a quicker turn around time could be expected resulting in both increased labor-savings and increased specimen throughput. As an added benefit, because the samples need not be machined, the penetration cavity remains intact and in its original state. This should result in a more accurate account of the projectile/armor interaction.

Since the system may be easily automated for data acquisition an additional increase in sample throughput may be realized. This would would provide scientists with greater freedom of choice as to when the images may be studied. Because the images are stored on magnetic media they are much easier to archive and store. They may later be recalled to provide much of the same information as the original sample and in a much more convenient format. As techniques for image processing and armor analysis improve these same images may be reprocessed using the newer techniques, providing additional data without the need for sample and testing redundancy.

V. Results and Discussion

The following figures illustrate some of the results that were obtained during this program effort. In Figure 2 the image processor was used to obtain a top view of the projectile cavity. Using this image as a scaled guide, it is possible for the computer to perform a morphological analysis and feature extraction on successive images. This allows the user to obtain very accurate measurements of the various features captured by the imaging system of the cavity interior.

Figure 3 is an example of one of the many ways in which the image processor can be used to display the imaging data. In this technique the data is displayed as a 3 dimensional plot, with pixel location being shown along the X and Y axes and gray level intensity being displayed along the Z axis. Such a plot is very useful in locating very subtle changes in gray scale values as well as providing a basis for feature extraction.

Figure 4 shows a map of the interior of the projectile penetration cavity. In this figure the borescope angle of rotation is shown along the X axis from left to right. For demonstration purposes the increment of rotation was chosen to be 90 degrees. A much smaller rotation angle could have been chosen for a more detailed view of the cavity interior. The depth of borescope penetration into the cavity is depicted along the Y axis in increments of 1.0, 1.5, 3.0 cm from the sample surface. Here again the increments are arbitrary and are given only for demonstration purposes. For this sequence of images, the camera chosen provided a resolution of 512 x 512 pixels.

Figure 5 depicts a map of the interior of the projectile cavity. The layout is the same as that of Figure 3. However the camera chosen provides a resolution of 1024 x 1024 pixels per image. As a result the image quality and sensitivity to detail have dramatically improved. It is important however to keep in mind that the file size and relative computing power required to process this sequence of images has also increased. The Figure 4 image requires approx. 3 megabytes of storage while Figure 5 requires 12 megabytes of storage (images only, not including text).

Figure 6 is an exploded view of one of the borescope images at a depth of 1.0 cm and an angular rotation of 90 degrees. The camera chosen provided a resolution of 1024 x 1024 pixels.

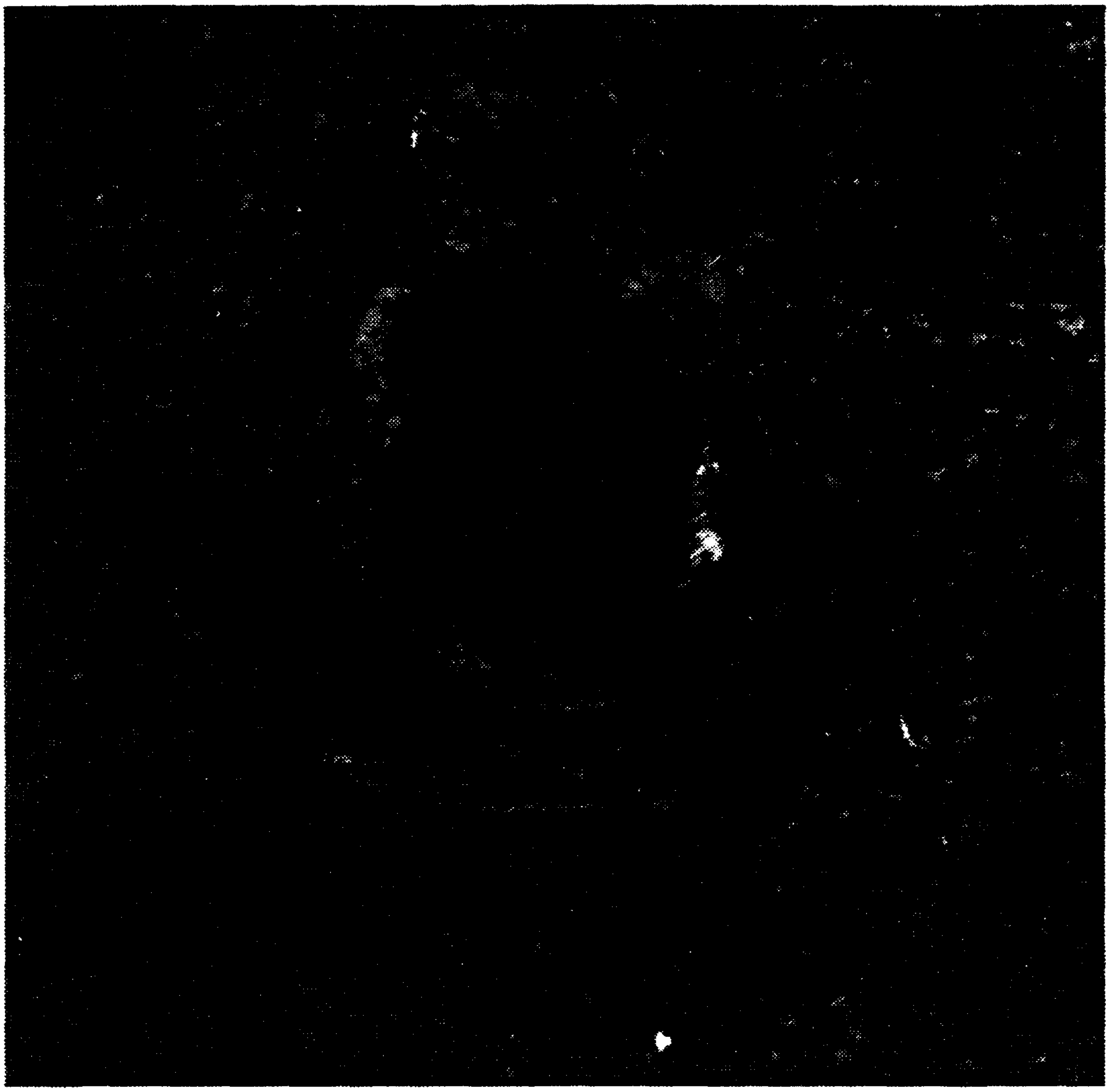


Figure 2. Exterior view of armor sample
Resolution: 1024x1024 pixels @ 8 bits/pixel

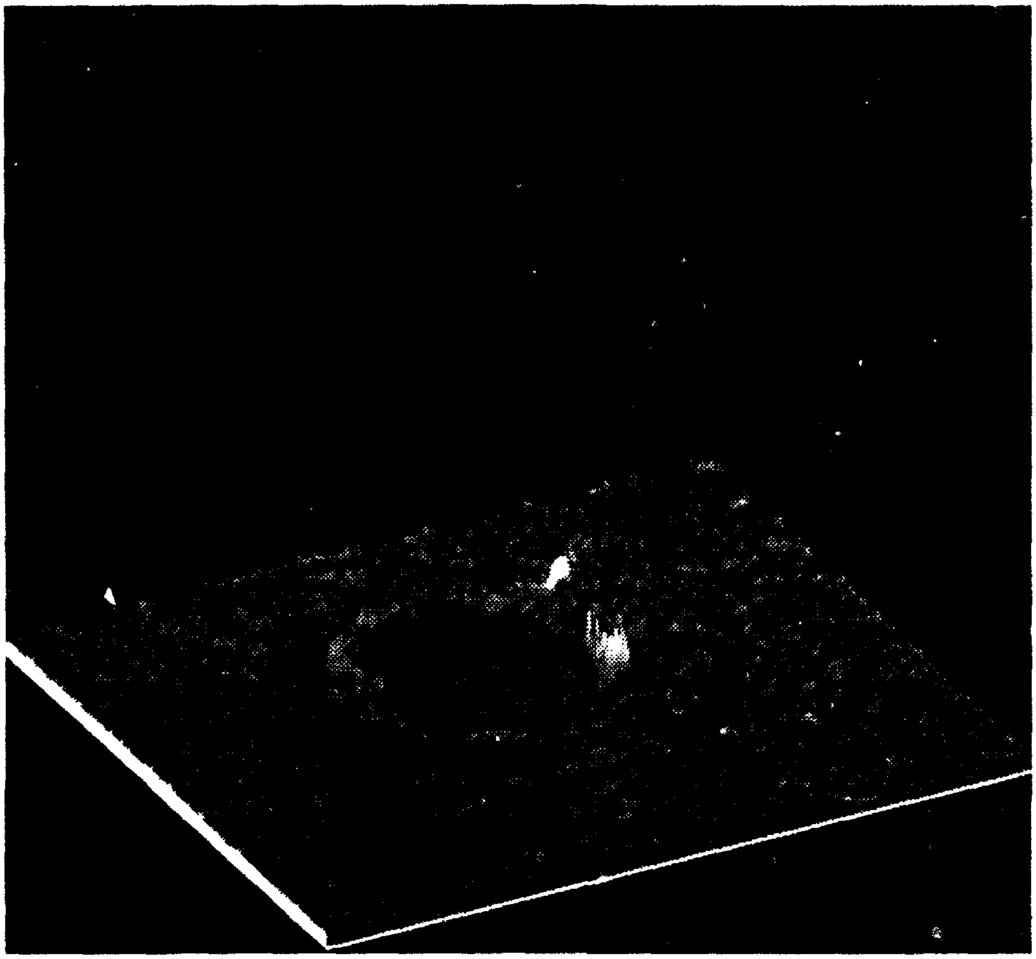
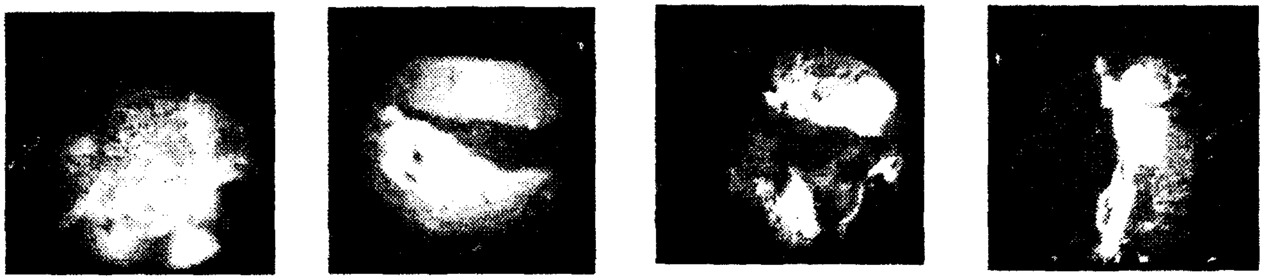


Figure 3. Exterior view of armor sample (3d)

Using 3d grey scale plot (Z axis)
Resolution: 1024x1024 @8 bits/pixel

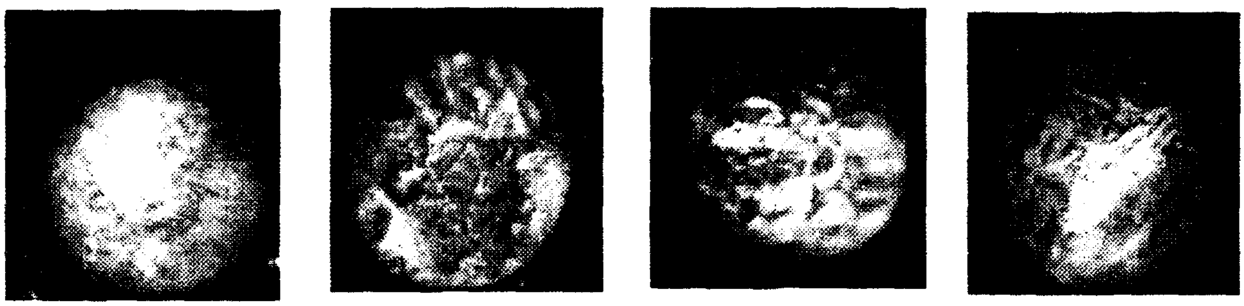
Angle of Rotation at depth of 1 cm.

0 Deg. 90 Deg. 180 Deg. 270 Deg.



Angle of Rotation at depth of 1.5 cm.

0 Deg. 90 Deg. 180 Deg. 270 Deg.



Angle of Rotation at depth of 3 cm.

0 Deg. 90 Deg. 180 Deg. 270 Deg.

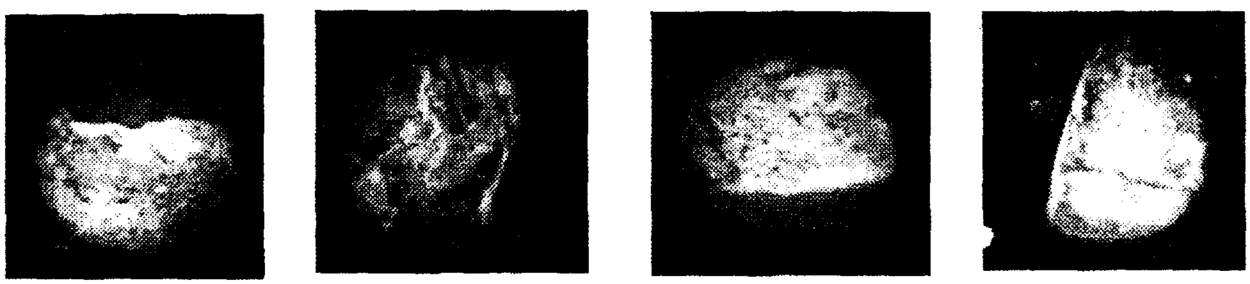
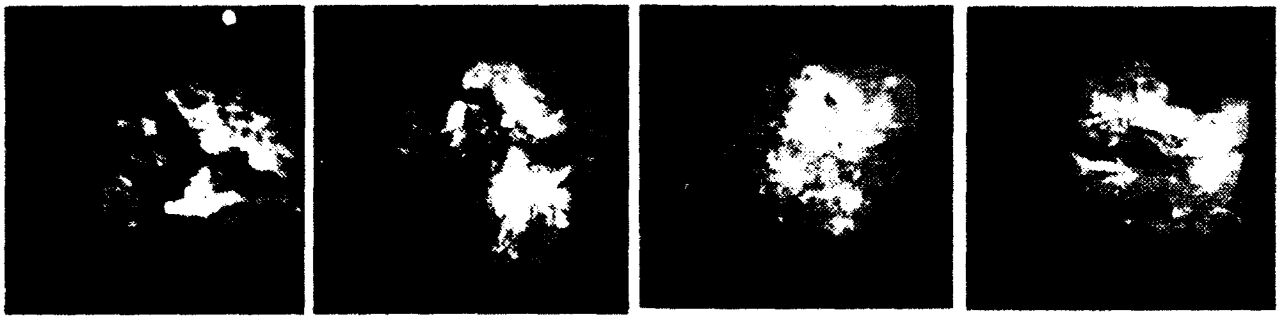


Figure 4. Interior cavity of ballistic sample.

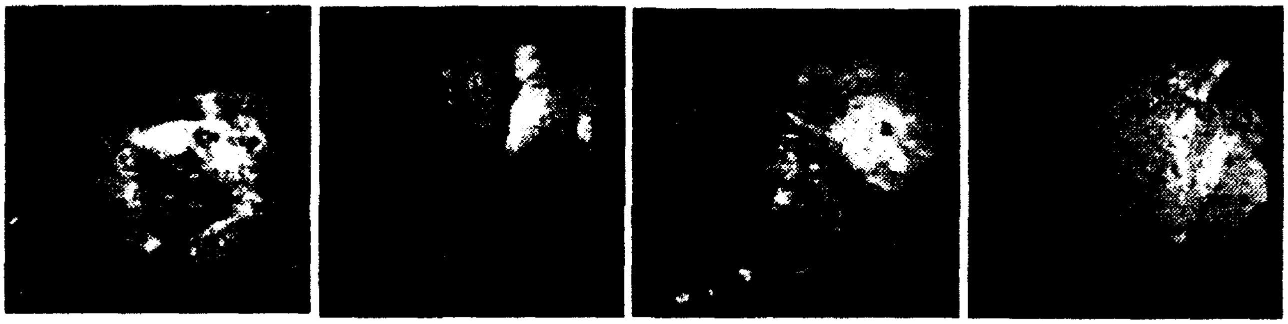
Angle of Rotation at depth of 1 cm.

0 Deg. 90 Deg. 180 Deg. 270 Deg.



Angle of Rotation at depth of 1.5 cm.

0 Deg. 90 Deg. 180 Deg. 270 Deg.



Angle of Rotation at depth of 3 cm.

0 Deg. 90 Deg. 180 Deg. 270 Deg.

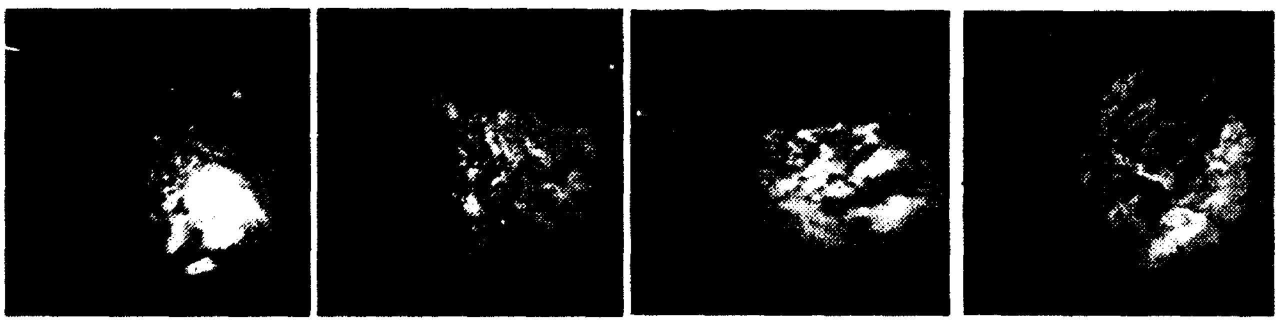


Figure 5. Interior cavity of ballistic sample. (high resolution)

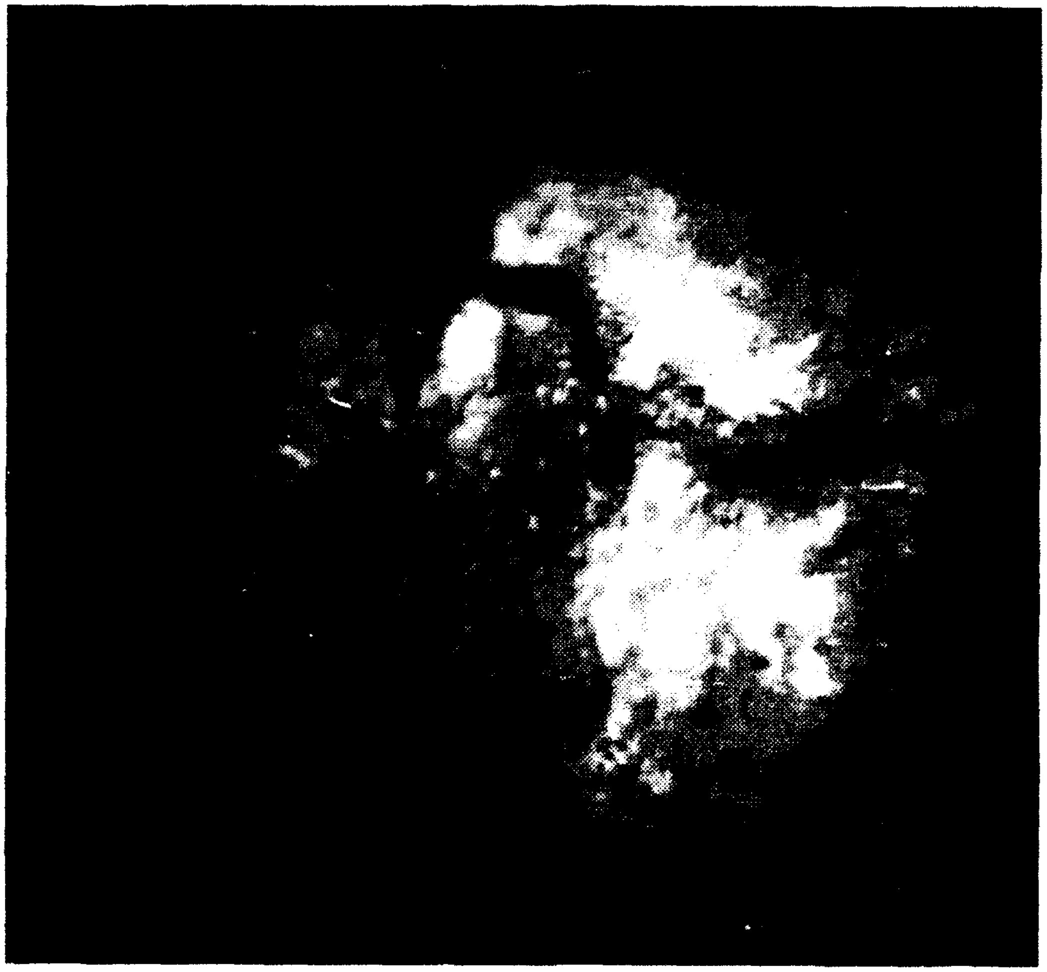


Figure 6. Interior view of armor sample

Depth of 1.0 cm., Borescopeangle of rotation 90 degrees
Resolution: 1024x1024 @8 bits/pixel

VI. Conclusions and Recommendations

Digital image processing technology has been employed to capture and analyze images of ballistic armor test specimens. Using digital image processing techniques to map out the interior of a ballistic penetration cavity, it is possible to identify the advent and or degree of projectile shatter in ballistic armor test specimens. With this type of information available, it becomes possible for the researcher to better select targets for inclusion into military specifications and standards. In addition, such a map may then be used by experts to gain additional information on approximate depth, location and degree of projectile shatter, possibly providing valuable information on projectile performance.

Using routines developed on the C-interpreter the digitized images, if so desired may be enhanced for greater detail and contrast. Digital image processing can provide reliable and quantitative information relating to visual changes induced in armor test specimens. Additional research is required to study the analytical relationships necessary to describe and quantify fracture characteristics and projectile/armor interaction. Finally, possible applications of this technology to other types of armor should be investigated (i.e. composites, metal, etc.)

It is recommended that a cooperative program be established between experts in the areas of armor, ballistics, and electronic imaging. Such a combined effort could provide a thorough study of the armor/projectile interaction utilizing digital image processing as the primary investigative tool. The end result would be the development of a low cost methodology for the study of armor/projectile interaction. Such a system could later be automated for increased productivity if necessary.

Appendix A

Micro Soft C Source Code
8 bit gray scale capture routine

```
/* This is a program that is used to capture a series of */
/* gray scale image files for display on the monitor in a tiled  */
/* fashion for easier visualization.                      */
/* This program was developed for the Armor Imaging project */
/* at the U.S. Army MTL research facility.                 */

#define AT
#include <try.h>
#include "c:\inc\stdlib.h"
#include "c:\inc\string.h"
#include "c:\inc\itex150.h"
#include "c:\inc\regop.h"
#define clrscr() printf("%c%c%c%c", 27, 91, 50, 74)

main ()
{

        /*Varaible declarations*/
        char fname[25],flabel[9],nimgchr[1],nlevelchr[1];
        char dir[9],selimgchr[2],response;
        int count,saoi[4],daoi[4],numimg,numlevel,selimg;
        float x,y,scaleval,deltay,deltax;
        /*Initialize system on next two lines*/
        load_cfg("");
        initsys();
        clrscr();
sprintf(dir,"d:\\armor\\");
select_path(B1);
saoi[0] = 0;
saoi[1] = 0;
saoi[2] = 512;
saoi[3] = 480;

printf("*** PLEASE MOUNT DRIVE D: (151 IMAGES) DISK *** \r\n");
printf("Please enter the Filename for the image series: \r\n");
scanf("%s",flabel);
printf("\r\n Please enter the number of images per level: ");
scanf("%s",nimgchr);
printf("\r\n Please enter the number of levels: ");
scanf("%s",nlevelchr);
numimg = atoi(nimgchr);
numlevel = atoi(nlevelchr);

deltay = (480.0/numlevel);
deltax = (512.0/numimg);
scaleval = (1.0/numimg);
count = 0;
```

```
grab(B1);
printf("Loading/Saving files To/From %s ...\r\n",dir);
for (y=1; y < 480.0; y+=deltay)
  {
  for (x=1; x < 512.0; x+=deltax)
    {
  count++;
    sprintf(fname,"%s%s%d.IMG",dir,flabel,count);
    printf("Please ready camera for image segment:\r\n");
    printf("Press ENTER when ready.....\r\n");
    response = getch();
    printf("Storing image segment to file %s%s ... \r\n",dir,fname);
    snap(B1);
    im_write(B1,EIGHT_BIT,0,0,512,512,fname);
    grab(B1);

    }
  }
    printf("Image capture complete. \r\n");
    clrscr();
end(0);
}
```

Micro Soft C Source Code
8 bit gray scale image display routine

```
/* This is a program that is used to display several */
/* series 151 image files on the monitor in a tiled  */
/* fashion for easier visualization.                 */
/* This program was developed for the Armor Imaging project */
/* at the U.S. Army MTL research facility.                  */


#define AT
#include <try.h>
#include "c:\inc\stdlib.h"
#include "c:\inc\string.h"
#include "c:\inc\itex150.h"
#include "c:\inc\regop.h"
#define clrscr() printf("%c%c%c%c", 27, 91, 50, 74)

main ()
{

          /*Varaible declarations*/
        char fname[25],flabel[9],nimgchr[1],nlevelchr[1];
        char dir[9],selimgchr[2],response;
        int count,saoi[4],daoi[4],numimg,numlevel,selimg;
        float x,y,scaleval,deltay,deltax;
          /*Initialize system on next two lines*/
          load_cfg("");
          initsys();
        clrscr();

sprintf(dir,"d:\\armor\\");
select_path(ALOW);
saoi[0] = 0;
saoi[1] = 0;
saoi[2] = 512;
saoi[3] = 480;

printf("*** PLEASE MOUNT DRIVE D: (151 IMAGES) DISK *** \r\n");
printf("Please enter the sequential Filename header: \r\n");
scanf("%s",flabel);
printf("\r\n Please enter the number of images per level: ");
scanf("%s",nimgchr);
printf("\r\n Please enter the number of levels: ");
scanf("%s",nlevelchr);
numimg = atoi(nimgchr);
numlevel = atoi(nlevelchr);

deltay = (480.0/numlevel);
deltax = (512.0/numimg);
```

```
scaleval = (1.0/numimg);
count = 0;

printf("Loading/Saving files To/From %s ...\r\n",dir);
for (y=1; y < 480.0; y+=deltay)
  {
  for (x=1; x < 512.0; x+=deltax)
    {
    count++;
    sprintf(fname,"%s%s%d.IMG",dir,flabel,count);
    printf("Loading image file %s ... \r\n",fname);
    im_read(B1,0,0 512,512,fname);
    daoi[0] = x;
    daoi[1] = y;
    daoi[2] = deltax;
    daoi[3] = deltay;
    scale(B1,ALOW,scaleval,scaleval,saoi,daoi);
    }
  }
   response = ' ';
   do
    {
     select_path(ALOW);
     printf("\r\n Please enter the number of an image for full scale view:
");
     scanf("%s",selimgchr);
     selimg = atoi(selimgchr);
     sprintf(fname,"%s%s%d.IMG",dir,flabel,selimg);
     im_read(B1,0,0,512,512,fname);
     select_path(B1);
     printf("\r\n Please press ENTER to Return to image map or Q to Quit: ");
     response = getch();
    }
   while (response != 'Q');
   load_cfg("");
   initsys();
   clrscr();
end(0);
}
```

Micro Soft C Source Code
8 Bit Grey Scale Mapping Program
for building image file Name.map

```
/* This is a program that is used to create a Grey Scale image map */
/* series 151 image file. This file may then be used to show*/
/* the borescope data in a tiled fashion for easier visualization. */
/* This program was developed for the Armor Imaging project */
/* at the U.S. Army MTL research facility.                  */

#define AT
#include <try.h>
#include "c:\inc\stdlib.h"
#include "c:\inc\string.h"
#include "c:\inc\itex150.h"
#include "c:\inc\regop.h"
#define clrscr() printf("%c%c%c%c", 27, 91, 50, 74)

main ()
{

          /*Varaible declarations*/
         char dir[9],Mname[25],fname[25],flabel[9],fmap[9];
         char nimgchr[1],nlevelchr[1],selimgchr[2],response;
         int count,saoi[4],daoi[4],numimg,numlevel,selimg;
         float x,y,scaleval,deltay,deltax;
          /*Initialize system on next two lines*/
          load_cfg("");
          initsys();
         clrscr();

sprintf(dir,"d:\\armor\\");
select_path(ALOW);
saoi[0] = 0;
saoi[1] = 0;
saoi[2] = 512;
saoi[3] = 480;
printf("*** PLEASE MOUNT DRIVE D: (151 IMAGES) DISK *** \r\n");
printf("Please enter the sequential Filename header: \r\n");
scanf("%s",flabel);
printf("Please enter the MAP Filename to be created: \r\n");
scanf("%s",fmap);
printf("\r\n Please enter the number of images per level: ");
scanf("%s",nimgchr);
printf("\r\n Please enter the number of levels: ");
scanf("%s",nlevelchr);

numimg = atoi(nimgchr);
numlevel = atoi(nlevelchr);
```

```
deltay = (480.0/numlevel);
deltax = (512.0/numimg);
scaleval = (1.0/numimg);
count = 0;
printf("Loading/Saving files To/From %s ... \r\n",dir);

for (y=1; y.< 480.0; y+=deltay)
  {
  for (x=1; x < 512.0; x+=deltax)
    {
    count++;
    sprintf(fname,"%s%s%d.IMG",dir,flabel,count);
    printf("Loading image file %s ... \r\n",fname);
    im_read(B1,0,0,512,512,fname);
    daoi[0] = x;
    daoi[1] = y;
    daoi[2] = deltax;
    daoi[3] = deltay;
    scale(B1,ALOW,scaleval,scaleval,saoi,daoi);
    }
   }
   sprintf(Mname,"%s%s.IMG",dir,fmap);
   printf("Writing image MAP file %s ... \r\n",Mname);
   im_write(ALOW,EIGHT_BIT,0,0,512,512,Mname);

   load_cfg("");
   initsys();
   clrscr();
end(0);
}
```

Micro Soft C Source Code
24 bit RGB color image display routine

```
/* This is a program that is used to display several RGB */
/* series 151 image files on the monitor in a tiled     */
/* fashion for easier visualization.                    */
/* This program was developed for the Armor Imaging project */
/* at the U.S. Army MTL research facility.                  */



#define AT
#include <try.h>
#include "c:\inc\stdlib.h"
#include "c:\inc\string.h"
#include "c:\inc\itex150.h"
#include "c:\inc\regop.h"
#define clrscr() printf("%c%c%c%c", 27, 91, 50, 74)

main ()
{

          /*Varaible declarations*/
          char
fname[15],flabel[6],fmap[6],nimgchr[1],nlevelchr[1],selimgchr[2],response;
          int count,saoi[4],daoi[4],numimg,numlevel,selimg;
          float x,y,scaleval,deltay,deltax;
          /*Initialize system on next two lines*/
          load_cfg("");
          initsys();
          clrscr();

printf("This program is designed to use a RGB color file sequence: \r\n");
printf("This sequence should consist of a RGB map image, \r\n");
printf("As well as the individual RGB Borescope images. \r\n");
printf("Please enter the sequential Filename header: \r\n");
scanf("%s",flabel);
printf("Please enter the corresponding RGB Map File Series Name: \r\n");
scanf("%s",fmap);

   response = ' ';
   do
    {
    sprintf(fname,"R%s.IMG",fmap);
    printf("Loading MAP image file %s ... \r\n",fname);
    im_read(ALOW,0,0,512,512,fname);

    sprintf(fname,"G%s.IMG",fmap);
    printf("Loading MAP image file %s ... \r\n",fname);
    im_read(AHIGH,0,0,512,512,fname);
```

```
    sprintf(fname,"B%s.IMG",fmap);
    printf("Loading MAP image file %s ... \r\n",fname);
    im_read(B1,0,0,512,512,fname);

    dyn_cps(0,0,512,512,VDC,VDAL,VDAH,VDB);
    fb_access(NONE,SCAN,SCAN);
    adi_cpsmode(DYNAMIC);

     printf("\r\n Please enter the number of an image for full scale view:
");
     scanf("%s",selimgchr);
     selimg = atoi(selimgchr);
     sprintf(fname,"R%s%d.IMG",flabel,selimg);
    printf("Loading image file %s ... \r\n",fname);
    im_read(ALOW,0,0,512,512,fname);
    sprintf(fname,"G%s%d.IMG",flabel,selimg);
    printf("Loading image file %s ... \r\n",fname);
    im_read(AHIGH,0,0,512,512,fname);
    sprintf(fname,"B%s%d.IMG",flabel,selimg);
    printf("Loading image file %s ... \r\n",fname);
    im_read(B1,0,0,512,512,fname);
    dyn_cps(0,0,512,512,VDC,VDAL,VDAH,VDB);
    fb_access(NONE,SCAN,SCAN);
    adi_cpsmode(DYNAMIC);

     printf("\r\n Please press ENTER to Return to image map or Q to Quit: ");
     response = getch();
    }
   while (response != 'Q');
   load_cfg("");
   initsys();
end(0);
}
```

DISTRIBUTION LIST

No. of Copies	To
1	Office of the Under Secretary of Defense for Research and Engineering, The Pentagon, Washington, DC 20301
	Director, U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783-1197
1	ATTN: AMSRL-OP-CI-AD, Technical Publishing Branch
1	AMSRL-OP-CI-AD, Records Management Administrator
	Commander, Defense Technical Information Center, Cameron Station, Building 5, 5010 Duke Street, Alexandria, VA 22304-6145
2	ATTN: DTIC-FDAC
1	MIA/CINDAS, Purdue University, 2595 Yeager Road, West Lafayette, IN 47905
	Commander, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709-2211
1	ATTN: Information Processing Office
	Commander, U.S. Army Materiel Command, 5001 Eisenhower Avenue, Alexandria, VA 22333
1	ATTN: AMCSCI
	Commander, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD 21005
1	ATTN: AMXSY-MP, H. Cohen
	Commander, U.S. Army Missile Command, Redstone Arsenal, AL 35809
1	ATTN: AMSMI-RD-CS-R/Doc
	Commander, U.S. Army Armament, Munitions and Chemical Command, Dover, NJ 07801
2	ATTN: Technical Library
	Commander, U.S. Army Natick Research, Development and Engineering Center, Natick, MA 01760-5010
1	ATTN: Technical Library
	Commander, U.S. Army Satellite Communications Agency, Fort Monmouth, NJ 07703
1	ATTN: Technical Document Center
	Commander, U.S. Army Tank-Automotive Command, Warren, MI 48397-5000
1	ATTN: AMSTA-ZSK
1	AMSTA-TSL, Technical Library
	Commander, White Sands Missile Range, NM 88002
1	ATTN: STEWS-WS-VT
	President, Airborne, Electronics and Special Warfare Board, Fort Bragg, NC 28307
1	ATTN: Library
	Director, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD 21005
1	ATTN: AMSRL-WT
	Commander, Dugway Proving Ground, UT 84022
1	ATTN: Technical Library, Technical Information Division
	Commander, U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783
1	ATTN: AMSRL-SS
	Director, Benet Weapons Laboratory, LCWSL, USA AMCCOM, Watervliet, NY 12189
1	ATTN: AMSMC-LCB-TL
1	AMSMC-LCB-R
1	AMSMC-LCB-RM
1	AMSMC-LCB-RP
	Commander, U.S. Army Foreign Science and Technology Center, 220 7th Street, N.E., Charlottesville, VA 22901-5396
3	ATTN: AIFRTC, Applied Technologies Branch, Gerald Schlesinger
	Commander, U.S. Army Aeromedical Research Unit, P.O. Box 577, Fort Rucker, AL 36360
1	ATTN: Technical Library

No. of Copies	To
	U.S. Army Aviation Training Library, Fort Rucker, AL 36360
1	ATTN: Building 5906-5907
	Commander, U.S. Army Agency for Aviation Safety, Fort Rucker, AL 36362
1	ATTN: Technical Library
	Commander, Clarke Engineer School Library, 3202 Nebraska Ave., N, Ft. Leonard Wood, MO 65473-5000
1	ATTN: Library
	Commander, U.S. Army Engineer Waterways Experiment Station, P.O. Box 631, Vicksburg, MS 39180
1	ATTN: Research Center Library
	Commandant, U.S. Army Quartermaster School, Fort Lee, VA 23801
1	ATTN: Quartermaster School Library
	Naval Research Laboratory, Washington, DC 20375
1	ATTN: Code 5830
2	Dr. G. R. Yoder - Code 6384
	Chief of Naval Research, Arlington, VA 22217
1	ATTN: Code 471
	Commander, U.S. Air Force Wright Research & Development Center, Wright-Patterson Air Force Base, OH 45433-6523
1	ATTN: WRDC/MLLP, M. Forney, Jr.
1	WRDC/MLBC, Mr. Stanley Schulman
	NASA - Marshall Space Flight Center, MSFC, AL 35812
1	ATTN: Mr. Paul Schuerer/EH01
	U.S. Department of Commerce, National Institute of Standards and Technology, Gaitherburg, MD 20899
1	ATTN: Stephen M. Hsu, Chief, Ceramics Division, Institute for Materials Science and Engineering
1	Committee on Marine Structures, Marine Board, National Research Council, 2101 Constitution Avenue, N.W., Washington, DC 20418
1	Materials Sciences Corporation, Suite 250, 500 Office Center Drive, Fort Washington, PA 19034
1	Charles Stark Draper Laboratory, 555 Technology Square, Cambridge, MA 02139
	Wyman-Gordon Company, Worcester, MA 01601
1	ATTN: Technical Library
	General Dynamics, Convair Aerospace Division P.O. Box 748, Fort Worth, TX 76101
1	ATTN: Mfg. Engineering Technical Library
	Plastics Technical Evaluation Center, PLASTEC, ARDEC Bldg. 355N, Picatinny Arsenal, NJ 07806-5000
1	ATTN: Harry Pebly
1	Department of the Army, Aerostructures Directorate, MS-266, U.S. Army Aviation R&T Activity - AVSCOM, Langley Research Center, Hampton, VA 23665-5225
1	NASA - Langley Research Center, Hampton, VA 23665-5225
	U.S. Army Vehicle Propulsion Directorate, NASA Lewis Research Center, 2100 Brookpark Road, Cleveland, OH 44135-3191
1	ATTN: AMSRL-VP
1	NASA - Lewis Research Center, 2100 Brookpark Road, Cleveland, OH 44135-3191
	Director, Defense Intelligence Agency, Washington, DC 20340-6053
1	ATTN: ODT-5A (Mr. Frank Jaeger)
	Director, U.S. Army Research Laboratory, Watertown, MA 02172-0001
2	ATTN: AMSRL-OP-CI-D, Technical Library
5	Author